УДК 519.1

Н.Н. ВОДОЛАЗОВ

ОБ ОСОБЕННОСТЯХ ПОТОКА В СЕТЯХ С БАРЬЕРНОЙ ДОСТИЖИМОСТЬЮ

Рассматривается задача о нахождении максимального потока в сетях с барьерной достижимостью при помощи модифицированного алгоритма Эдмондса-Карпа. Особенность предложенного алгоритма в том, что он позволяет находить поток в сетях, для которых разработанный ранее алгоритм [1] не находил решения.
Ключевые слова: *сеть, источник, сток, поток, блокирующий поток.*

Введение. В классической постановке задачи стационарным потоком на графе $G(U,V,\varphi)$ называется отображение $f:U \to R^{+}$, удовлетворяющее условиям:

$$\begin{cases} \sum\limits_{u\in\Theta_{-}(\upsilon)} f(u) = \sum\limits_{u\in\Theta_{+}(\upsilon)} f(u), \quad \forall\, \upsilon \in V, \upsilon \neq s, \upsilon \neq t, \\ 0 \leq f(u) \leq c(u), \end{cases} \quad (1)$$

где $\Theta_{-}(\upsilon)$ – множество дуг, входящих в вершину υ; $\Theta_{+}(\upsilon)$ – множество дуг, выходящих из вершины υ; $c(u)$ – пропускная способность дуги; s – источник; t – сток.

Требуется максимизировать $\sum\limits_{u\in\Theta_{-}(s)} f(u)$.

Для решения этой задачи может быть применен симплекс-метод. Однако в 1956 г. Форд и Фалкерсон предложили более эффективный алгоритм решения задачи, основанный на поиске увеличивающих цепей из источника в сток. Эдмондс и Карп [2] в 1972 г. улучшили алгоритм Форда-Фалкерсона и получили алгоритм, время работы которого не зависело от пропускных способностей дуг.

Постановка задачи. Требуется найти максимальный поток в графе $G(U,V,\varphi)$ с дополнительным ограничением на пути частиц, которое в [1] названо барьерной достижимостью. Множество дуг графа U разбито на три непересекающихся подмножества: U_{H} (нейтральные дуги), U_{+} (увеличивающие дуги) и $U_{Б}$ (барьерные дуги). С каждой частицей, проходящей по графу, связано целое число i - её барьерный уровень. Для частиц, выходящих из источника, он равен 0. Если барьерный уровень меньше некоторого заданного числа k, то при прохождении дуг из множества U_{+} барьерный уровень увеличивается на 1, при прохождении дуг из множеств U_{H} и $U_{Б}$ барьерный уровень не изменяется. Частица может проходить по дугам множества $U_{Б}$ только в случае, если её барьерный уровень равен k.

Существующее решение. Решение описанной задачи предложено в [1]. Для этого по исходному графу строится вспомогательный граф. Каждой вершине x исходного графа ставится в соответствие $k+1$ вершина

вспомогательного графа $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$. Каждой нейтральной дуге (x, y) исходного графа ставится в соответствие $k+1$ дуга вспомогательного графа вида $(x^{(i)}, y^{(i)})$, $i = 0, \ldots, k$ (рис.1).

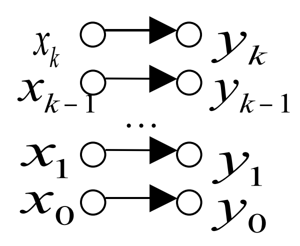


Рис.1. Преобразование нейтральных дуг

Каждой увеличивающей дуге исходного графа (x, y) ставится в соответствие $k+1$ дуга вспомогательного графа: k дуг вида $(x^{(i)}, y^{(i+1)})$, $i = 0, \ldots, k-1$ и одна дуга $(x^{(k)}, y^{(k)})$ (рис.2).

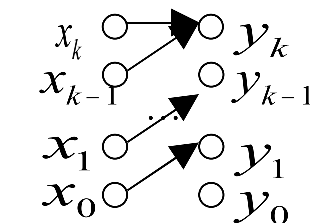


Рис.2. Преобразование увеличивающих дуг

Каждой барьерной дуге исходного графа ставится в соответствие одна дуга вспомогательного графа $(x^{(k)}, y^{(k)})$ (рис.3).

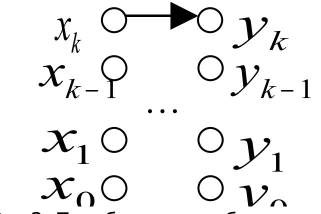


Рис.3. Преобразование барьерных дуг

В соответствии с идеей, изложенной в [1], пропускная способность дуг вспомогательного графа не определена. Вместо этого определен максимальный поток по множеству дуг вспомогательного графа, который поставлен в соответствие дуге исходного графа (множеству связанных дуг). Следовательно, применить алгоритм Форда-Фалкерсона мы не можем.

В работе [1] авторами был развит новый подход к построению алгоритма. Его сущность состоит в том, что алгоритм ищет кратчайший простой путь на вспомогательном графе из источника в сток, по которому можно пустить увеличивающий поток. Поток увеличивается на некоторую величину, и пропускная способность множеств связанных дуг, через которые прошел увеличивающий путь, уменьшается на эту же величину. Таким об-

разом, в отличие от алгоритма Форда-Фалкерсона алгоритм, описанный в [1], ищет не увеличивающую цепь, а увеличивающий путь. Ясно, что найденный поток будет неулучшаемым.

Приведенный ниже пример показывает, что неулучшаемый поток может быть не максимальным в случае, когда возможен блокирующий поток [3]. Блокирующий поток в [3] рассматривается как поток, для которого не существует увеличивающего пути из источника в сток. Надо сказать, в таких случаях (блокировки) алгоритм Форда-Фалкерсона находит выход, поскольку существует увеличивающая цепь. Рассмотрим пример.

Пример 1. Пусть дан граф G (рис.4).

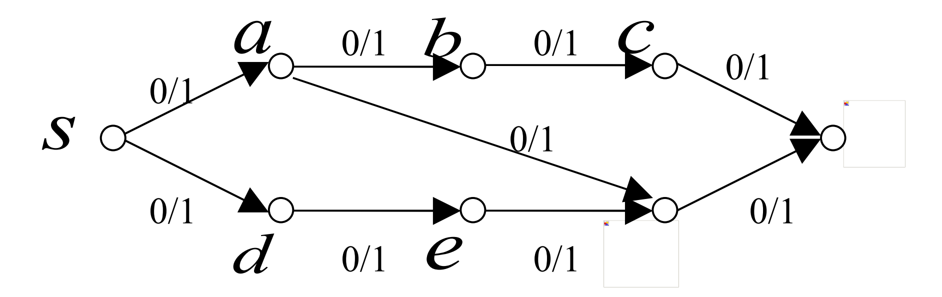


Рис.4. Нулевой поток на графе G

На рис. 4 источник – вершина s, сток – вершина t. Первая цифра в записи $0/1$ - поток по дуге, вторая – пропускная способность. Все дуги нейтральные и нет барьерных дуг, т.е. мы рассматриваем случай поиска потока в обычном графе. Тогда первый увеличивающий путь, найденный алгоритмом, будет проходить по вершинам s, a, f, t (рис.5).

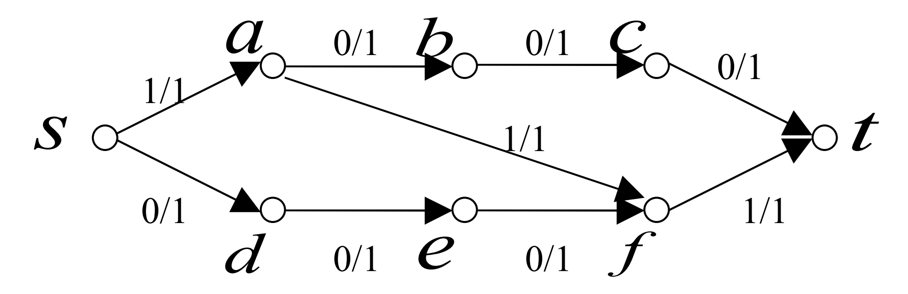


Рис.5. Блокирующий поток на графе G

Поток на рис. 5 – блокирующий, так как нет увеличивающего пути из источника в сток. Величина потока равна 1, максимальный поток, изображенный на рис. 6, равен 2.

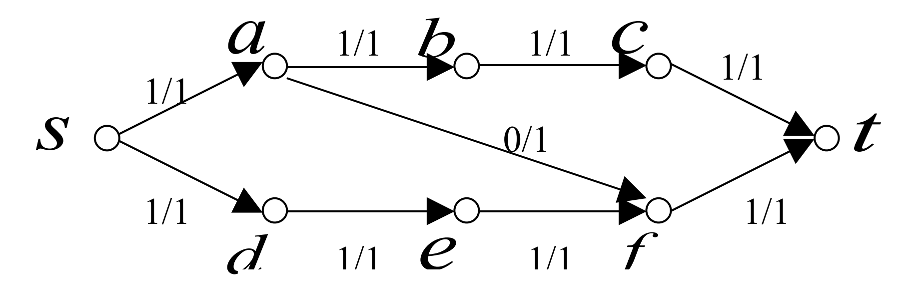


Рис.6. Максимальный поток на графе G

Таким образом алгоритм не находит максимальный поток для решения классической задачи. В случае с добавлением дополнительных ограничений на поток также возможно появление блокирующего потока.

Предлагаемый алгоритм решения задачи. Для удобства будем рассматривать не суммарный поток частиц всех барьерных уровней через дугу, а поток частиц каждого барьерного уровня через дугу. Поэтому для поставленной задачи полагаем, что поток – отображение $f : U \to (R^+)^{k+1}$, т.е. поток ставит каждой дуге в соответствие вектор чисел, при этом i -я компонента вектора является потоком i -го барьерного уровня по дуге. Обозначим $f_i(u,\upsilon)$ i -ю компоненту вектора потока по дуге (u,υ), а $f(u,\upsilon) = \sum_{i=0}^{k} f_i(u,\upsilon)$.

Условия (1) заменим на следующие условия:

$$\begin{cases} \sum\limits_{u\in\Theta_-(\upsilon)\cap U_H} f_0(u) = \sum\limits_{u\in\Theta_+(\upsilon)} f_0(u), \ \forall \upsilon \in V, \upsilon \neq s, \upsilon \neq t; \\ \sum\limits_{u\in\Theta_-(\upsilon)\cap U_H} f_i(u) + \sum\limits_{u\in\Theta_-(\upsilon)\cap U_+} f_{i-1}(u) = \sum\limits_{u\in\Theta_-(\upsilon)} f_i(u), \\ \forall i = 1,\ldots,k-1, \ \forall \upsilon \in V; \\ \sum\limits_{u\in\Theta_-(\upsilon)\cap U_H} f_k(u) + \sum\limits_{u\in\Theta_-(\upsilon)\cap U_+} f_{k-1}(u) + \sum\limits_{u\in\Theta_-(\upsilon)\cap U_+} f_k(u) + \sum\limits_{u\in\Theta_-(\upsilon)\cap U_Б} f_k(u) = \sum\limits_{u\in\Theta_-(\upsilon)} f_k(u) \\ \forall \upsilon \in V; \quad (2) \\ f_i(u) \geq 0, \ \forall i = 1,\ldots,k-1, \forall u \in U; \\ 0 \leq f(u) \leq c(u). \end{cases}$$

Требуется максимизировать $\sum\limits_{u\in\Theta_-(s)} f(u) - \sum\limits_{u\in\Theta_+(s)} f(u)$, т.е. вместо одного уравнения для каждой вершины в классическом случае, для данной задачи получается $k+1$ уравнение для каждой вершины. Следует подчеркнуть, что эту задачу так же, как и классическую, можно решать симплекс-методом, но мы попытаемся изменить алгоритм Эдмондса-Карпа так, чтобы с его помощью можно было решать задачу о максимальном потоке на графе с дополнительными ограничениями на пути частиц. Модифицировав алгоритм Форда-Фалкерсона, Эдмондс и Карп предложили производить увеличение потока по кратчайшей увеличивающей цепи (считая, что все дуги имеют единичную длину) [2].

Для решения поставленной задачи предполагается помечать каждую вершину вектором из $k+1$ метки, где k – максимальный барьерный уровень; каждая метка содержит информацию об увеличивающей цепи соответствующего барьерного уровня в данную вершину. Наш алгоритм на первом этапе решения использует в своей работе очередь помеченных вершин и вспомогательный алгоритм пометки вершин.

Алгоритм пометки вершин

На входе помеченная вершина x графа G. Алгоритм помечает смежные вершины и добавляет их в очередь помеченных вершин.

1. Для каждой дуги (x,a) и (a,x).
2. Для каждой метки (b,c,d) i-го уровня вершины x.
   a. Если мы рассматриваем дугу (x,a):
      i. если $f(x,a) < C(x,a)$, то обозначим $e = \min(d, C(x,a) - f(x,a))$, $s = "+"$, иначе переходим на шаг 2 и рассматриваем следующую метку.
   b. Если мы рассматриваем дугу (a,x):
      i. если $f_i(a,x) > 0$, то обозначим $e = \min(d, f_i(x,a))$, $s = "-"$, иначе переходим на шаг 2 и рассматриваем следующую метку.
   c. Если (x,a) или (a,x) – нейтральная дуга и a не помечена на i-ом уровне, то:
      i. помечаем вершину a на i-ом уровне меткой (x,s,e);
      ii. если a не первая в очереди, то добавляем a в очередь помеченных вершин.
   d. Если (x,a) – увеличивающая дуга, $i < k$ и a не помечена на $(i+1)$-ом уровне, то:
      i. помечаем вершину a на $(i+1)$-ом уровне меткой (x,s,e);
      ii. если a не первая в очереди, то добавляем a в очередь помеченных вершин.
   e. Если (a,x) – увеличивающая дуга, $i > 0$ и a не помечена на $(i-1)$-ом уровне, то:
      i. помечаем вершину a на $(i-1)$-ом уровне меткой (x,s,e);
      ii. если a не первая в очереди, то добавляем a в очередь помеченных вершин.
   f. Если (x,a) или (a,x) – барьерная дуга, $i = k$ и a не помечена на k-ом уровне, то:
      i. помечаем вершину a на k-ом уровне меткой (x,s,e);

ii. если a не первая в очереди, то добавляем a в очередь помеченных вершин.

Алгоритм поиска максимального потока

1. Поиск увеличивающей цепи.
   a. Помечаем источник вектором $\begin{pmatrix} (\sim, \sim, \infty) \\ - - - \\ - - - \\ \ldots \\ - - - \end{pmatrix}$, прочерк на i -ом месте означает, что вершина не помечена на данном барьерном уровне.
   b. Помещаем источник в очередь помеченных вершин.
   c. Пока очередь просмотренных вершин не пуста:
      i. выбираем одну вершину x из очереди;
      ii. выполняем для нее алгоритм пометки вершин, если сток оказался помечен, то найдена увеличивающая цепь, переходим к шагу 2, иначе переходим к шагу i;
      iii. если очередь просмотренных вершин пуста, то алгоритм завершает свою работу.
2. Поиск величины, на которую можно увеличить поток.
   a. Восстанавливаем увеличивающую цепь в сток по меткам. При этом цепь может проходить по некоторым дугам несколько раз, но на разных барьерных уровнях.
   b. Обозначим λ величину e из метки стока.
   c. Проходим по полученной цепи.
      i. Пусть l - количество проходов цепи по дуге, при котором направления цепи и дуги совпадают, p - количество проходов, при котором направления противоположны. Тогда, если $l > p$ и $\lambda > \dfrac{C(x,a) - f(x,a)}{l - p}$, то присвоим $\lambda = \dfrac{C(x,a) - f(x,a)}{l - p}$.
3. Увеличение потока. Проходим по всем дугам цепи:
   a. Если направление цепи из источника в сток совпадает с направлением дуги, то увеличиваем поток по дуге на величину λ .

b. Если направление дуги не совпадает с направлением цепи из источника в сток, то уменьшаем поток по дуге на величину λ .

Отличие предлагаемого нами алгоритма от алгоритма Форда-Фалкерсона в том, что, во-первых, он работает с вектором меток, а во-вторых, требуется шаг 2 для определения максимального потока по цепи. Необходимость шага 2 вызвана тем, что увеличивающая цепь может проходить по дуге несколько раз на разных барьерных уровнях. Благодаря использованию очереди каждая увеличивающая цепь будет иметь минимальную длину.

Пример 2. Рассмотрим граф с источником s и стоком t (рис.7), пропускные способности всех дуг равны 1. Обозначим поток через дугу вектором, первая компонента которого – поток с нулевым барьерным уровнем, вторая – с первым.

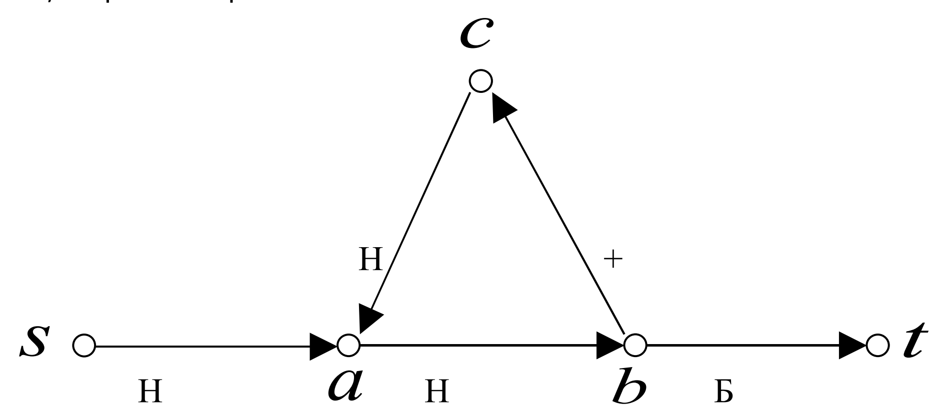


Рис.7. Начальное состояние графа

Дуга (b,t) – барьерная, с барьерным уровнем 1, дуга (b,c) – увеличивающая, остальные дуги нейтральные. Применим наш алгоритм.

После первого шага вершины окажутся помеченными так, как изображено на рис.8.

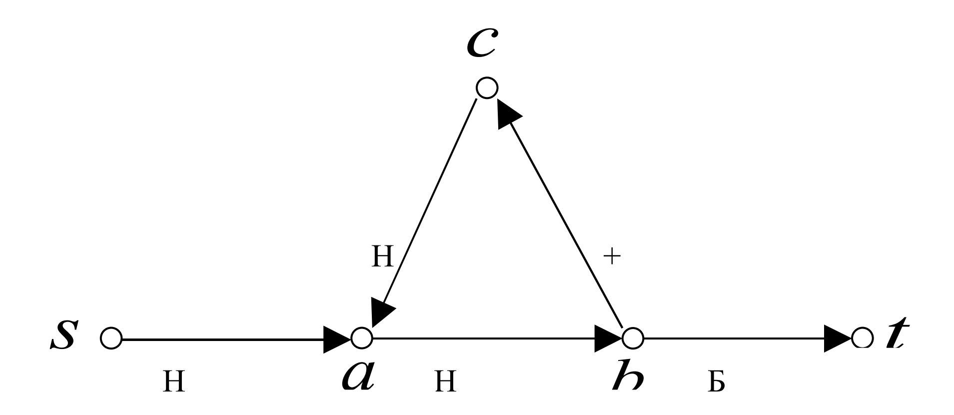


Рис. 8. Состояние графа после первого шага

Таким образом, найдена увеличивающая цепь из источника в сток, проходящая по вершинам s,a,b,c,a,b,t . При этом по дуге (a,b) цепь проходит дважды: сначала на первом барьерном уровне, затем на втором. Поэтому на втором шаге алгоритма величина, на которую по этой цепи можно увеличить поток, будет определена как $0{,}5$. После увеличения потока и повторения первого шага алгоритма граф будет выглядеть следующим образом (рис. 9).

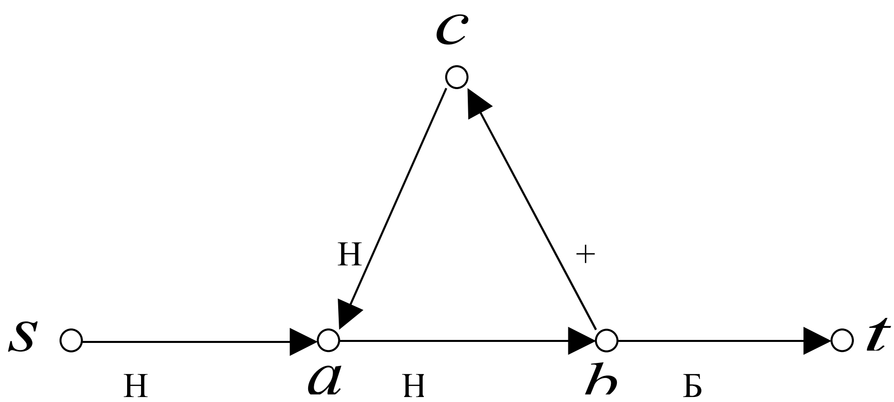


Рис. 9. Состояние графа, при котором алгоритм завершает работу

Суммарный поток по дуге (a,b) равен 1, и эта дуга насыщена. Вершина c не может быть помечена, так как вершина a помечена на нулевом уровне, а поток по дуге (c,a) идет по первому уровню. Пометка вершины c и построение через нее увеличивающей цепи означали бы нарушение условий (2). Сток не может быть помечен, и поэтому алгоритм заканчивает свою работу. На основе примера 2 поток, получаемый предложенным алгоритмом, совпадает с потоком, получаемым симплекс-методом для системы (3).

$$\begin{cases} f_0(s,a) \geq 0 \\ f_0(a,b) \geq 0 \\ f_1(a,b) \geq 0 \\ f_0(b,c) \geq 0 \\ f_1(b,c) \geq 0 \\ f_0(c,a) \geq 0 \\ f_1(c,a) \geq 0 \\ f_0(b,t) \geq 0 \\ f_1(b,t) \geq 0 \\ f_0(s,a) \leq 1 \\ f_0(a,b) + f_1(a,b) \leq 1 \\ f_0(b,c) + f_1(b,c) \leq 1 \\ f_0(c,a) + f_1(c,a) \leq 1 \\ f_0(b,t) + f_1(b,t) \leq 1 \\ f_0(s,a) + f_0(c,a) = f_0(a,b) \\ f_1(c,a) = f_1(a,b) \\ f_0(a,b) = f_0(b,c) \\ f_1(a,b) = f_1(b,c) + f_1(b,t) \\ f_0(c,a) = 0 \\ f_0(b,c) + f_1(b,c) = f_1(c,a) \\ f_0(s,a) \longrightarrow \max \end{cases} \tag{3}$$

Пример 3. Несмотря на то, что предлагаемый нами алгоритм может находить максимальный поток в сетях с возможной блокировкой, как в примере 1, существуют сети, для которых он не находит максимальное решение. В этом случае рассмотрим граф, изображенный на рис. 10.

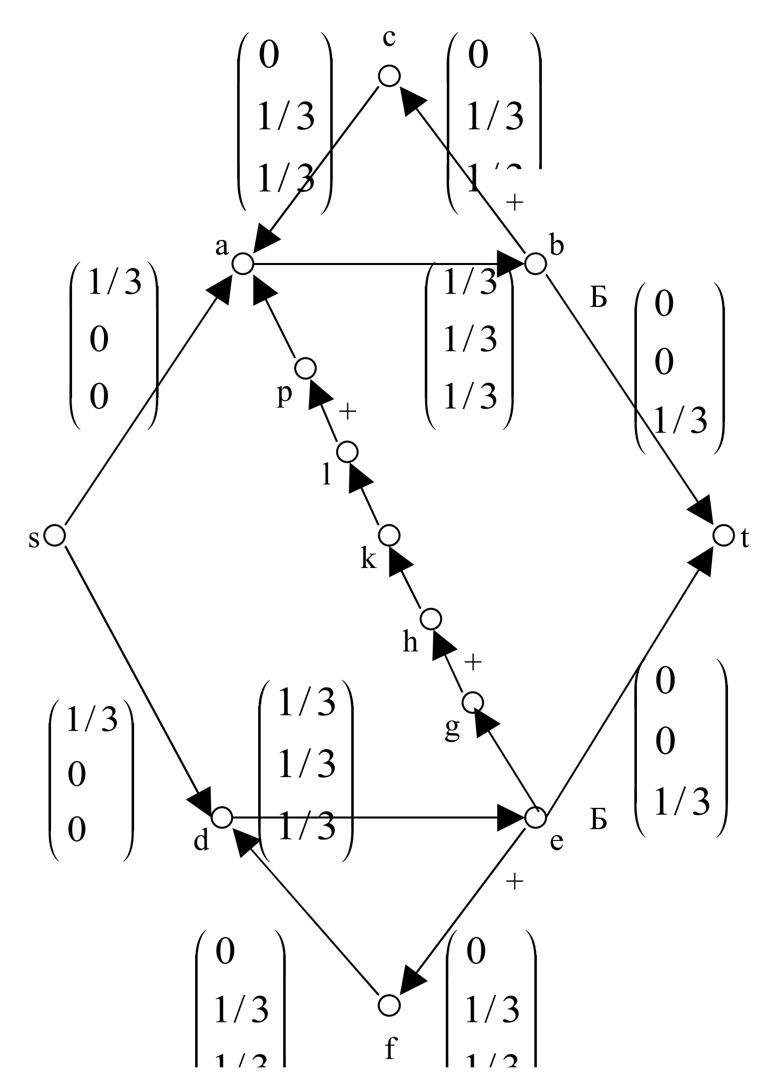


Рис.10. Граф с блокирующим потоком

Пропускные способности всех дуг равны 1. Дуги (b,c), (e,f), (g,h) и (l,p) - увеличивающие, (b,t) и (e,t) - барьерные, с барьерным уровнем 2, остальные дуги нейтральные. Поток по не помеченным на рис.10 дугам считаем равным 0, общий поток на рис.10 равен $\frac{2}{3}$.

Установлено, что найденный симплекс-методом максимальный поток, равен 1. Поток образуется, если все дуги пути, проходящего по вершинам s,d,e,g,h,k,l,p,a,b,t, насыщены, а остальные дуги не нагружены. Так как на рис. 10 дуги (d,e) и (a,b) насыщены, а (c,a), (f,d) и (p,a) не нагружены на нулевом барьерном уровне, то не существует не только увеличивающего пути, но и увеличивающей цепи из источника в сток. Следовательно, предложенный алгоритм не может найти максимальный поток на этом графе.

Выводы. Предложенный нами алгоритм основан на идее Форда-Фалкерсона нахождения увеличивающих цепей. Ранее нами было показано, что для поставленной задачи с барьерными ограничениями существуют блокирующие потоки – потоки, которые не максимальны, но для них не существует увеличивающей цепи. Из примера 3 видно, что алгоритм, основанный на поиске увеличивающей цепи, не может найти максимальный поток в сети на рис. 10. Отличие данной блокировки от блокировки, рассмотренной в [2], состоит в том, что блокировка в [2] представлена как отсутствие увеличивающего пути на графе без дополнительных ограничений на пути частиц. Тем не менее, решать эту задачу возможно, применяя симплекс-метод к системе уравнений и ограничений (2).

Автор выражает благодарность профессору Я.М.Ерусалимскому за постановку задачи.

Библиографический список

1. *Ерусалимский Я.М., Скороходов В.А.* Потоки в сетях со связанными дугами./ Я.М.Ерусалимский, В.А.Скороходов. // Изв. вузов. Сев.-Кав. регион. Естественные науки, 2003; прил. № 8. – С. 9-12.
2. Edmonds Jack, Karp Richard M. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems.//Journal of the Association for Computing Machinery. - Vol. 19. – No. 2, April 1972, –P. 248-264.
3. Ning Xuanxi, Ning Angelika The Blocking Flow Theory and its Application to Hamiltonian Graph Problems. Germany,Aachen: Shaker Verlag GmbH, 2006. –249p.

Материал поступил в редакцию 17.05.08.

N.N. VODOLAZOV

ON THE FLOW PECULIARITY
IN NETWORK WITH BARRIER REACHABILITY

The problem of finding maximal flow in networks with barrier reachability is considered. It is shown that existing algorithm for solution this problem can’t find solution for all networks. New algorithm is suggested.

ВОДОЛАЗОВ Николай Николаевич (р. 1984), аспирант первого года обучения факультета математики, механики и компьютерных наук Южного федерального университета. Окончил ЮФУ (2006), магистратуру (2007).
Научные интересы: теория графов и её приложения.